

In the Specification:

Please amend the title as follows:

PIPELINE ACCELERATOR THAT RECEIVES OR TRANSMITS DATA IN A MESSAGE, FOR IMPROVED COMPUTING ARCHITECTURE AND RELATED SYSTEM AND METHOD

Please amend the specification as follows:

[2] This application is related to U.S. Patent App. Serial Nos. 10/684,102 entitled IMPROVED COMPUTING ARCHITECTURE AND RELATED SYSTEM AND METHOD (~~Attorney Docket No. 1934-11-3~~), 10/684,053 entitled COMPUTING MACHINE HAVING IMPROVED COMPUTING ARCHITECTURE AND RELATED SYSTEM AND METHOD (~~Attorney Docket No. 1934-12-3~~), 10/684,057 entitled PROGRAMMABLE CIRCUIT AND RELATED COMPUTING MACHINE AND METHOD (~~Attorney Docket No. 1934-14-3~~), and 10/683,932 entitled PIPELINE ACCELERATOR HAVING MULTIPLE PIPELINE UNITS AND RELATED COMPUTING MACHINE AND METHOD (~~Attorney Docket No. 1934-15-3~~), which have a common filing date and owner and which are incorporated by reference.

[54] The pipeline accelerator **44** is disposed on at least one PLIC (not shown) and includes hardwired pipelines **74₁ – 74_n**, which process respective data without executing program instructions. The firmware memory **52** stores the configuration firmware for the accelerator **44**. If the accelerator **44** is disposed on multiple PLICs, these PLICs and their respective firmware memories may be disposed in multiple pipeline units (**FIG. 4**). The accelerator **44** and pipeline units are discussed further below and in previously cited U.S. Patent App. Serial No. 10/683,932 entitled PIPELINE ACCELERATOR HAVING MULTIPLE PIPELINE UNITS AND RELATED COMPUTING MACHINE AND METHOD (~~Attorney Docket No. 1934-15-3~~).

Alternatively, the accelerator **44** may be disposed on at least one ASIC, and thus may have internal interconnections that are unconfigurable. In this alternative, the machine

40 may omit the firmware memory **52**. Furthermore, although the accelerator **44** is shown including multiple pipelines **74**, it may include only a single pipeline. In addition, although not shown, the accelerator **44** may include one or more processors such as a digital-signal processor (DSP). Moreover, although not shown, the accelerator **44** may include a data input port and/or a data output port.

[55] The general operation of the peer-vector machine **40** is discussed in previously cited U.S. Patent App. Serial No. 10/684,102 entitled IMPROVED COMPUTING ARCHITECTURE AND RELATED SYSTEM AND METHOD (~~Attorney Docket No. 1934-11-3~~), and the structure and operation of the pipeline accelerator **44** is discussed below in conjunction with **FIGS. 4 – 9**.

[57] The accelerator **44** includes one or more pipeline units **78**, each of which includes a pipeline circuit **80**, such as a PLIC or an ASIC. As discussed further below and in previously cited U.S. Patent App. Serial No. 10/683,932 entitled PIPELINE ACCELERATOR HAVING MULTIPLE PIPELINE UNITS AND RELATED COMPUTING MACHINE AND METHOD (~~Attorney Docket No. 1934-15-3~~), each pipeline unit **78** is a “peer” of the host processor **42** and of the other pipeline units of the accelerator **44**. That is, each pipeline unit **78** can communicate directly with the host processor **42** or with any other pipeline unit. Thus, this peer-vector architecture prevents data “bottlenecks” that otherwise might occur if all of the pipeline units **78** communicated through a central location such as a master pipeline unit (not shown) or the host processor **42**. Furthermore, it allows one to add or remove peers from the peer-vector machine **40** (**FIG. 3**) without significant modifications to the machine.

[61] The hardwired pipelines **74₁-74_n** perform respective operations on data as discussed above in conjunction with **FIG. 3** and in previously cited U.S. Patent App. Serial No. 10/684,102 entitled IMPROVED COMPUTING ARCHITECTURE AND RELATED SYSTEM AND METHOD (~~Attorney Docket No. 1934-11-3~~), and the communication shell **84** interfaces the pipelines to the other components of the pipeline

circuit **80** and to circuits (such as a data memory **92** discussed below) external to the pipeline circuit.

[64] The configuration manager **90** sets the soft configuration of the hardwired pipelines **74₁-74_n**, the communication interface **82**, the communication shell **84**, the controller **86**, the exception manager **88**, and the interface **91** in response to soft-configuration data from the host processor **42** (**FIG. 3**) — as discussed in previously cited U.S. Patent App. Serial No. 10/684,102 — entitled IMPROVED COMPUTING ARCHITECTURE AND RELATED SYSTEM AND METHOD (~~Attorney Docket No. 1934-41-3~~), the hard configuration denotes the actual topology, on the transistor and circuit-block level, of the pipeline circuit **80**, and the soft configuration denotes the physical parameters (e.g., data width, table size) of the hard-configured components. That is, soft configuration data is similar to the data that can be loaded into a register of a processor (not shown in **FIG. 4**) to set the operating mode (e.g., burst-memory mode) of the processor. For example, the host processor **42** may send soft-configuration data that causes the configuration manager **90** to set the number and respective priority levels of queues in the communication interface **82**. The exception manager **88** may also send soft-configuration data that causes the configuration manager **90** to, e.g., increase the size of an overflowing buffer in the communication interface **82**.

[68] As discussed above in conjunction with **FIG. 3**, where the pipeline circuit **80** is a PLIC, the firmware memory **52** stores the firmware that sets the hard configuration of the pipeline circuit. The memory **52** loads the firmware into the pipeline circuit **80** during the configuration of the accelerator **44**, and may receive modified firmware from the host processor **42** (**FIG. 3**) via the communication interface **82** during or after the configuration of the accelerator. The loading and receiving of firmware is further discussed in previously cited U.S. Patent App. Serial No. 10/684,057 — entitled PROGRAMMABLE CIRCUIT AND RELATED COMPUTING MACHINE AND METHOD (~~Attorney Docket No. 1934-44-3~~).

[87] The controller **86** includes a sequence manager **148** and a synchronization interface **150**, which receives one or more synchronization signals SYNC. A peer, such as the host processor **42** (**FIG. 3**), or a device (not shown) external to the peer-vector machine **40** (**FIG. 3**) may generate the SYNC signal, which triggers the sequence manager **148** to activate the hardwired pipelines **74₁-74_n** as discussed below and in previously cited U.S. Patent App. Serial No. 10/683,932 entitled PIPELINE ACCELERATOR HAVING MULTIPLE PIPELINE UNITS AND RELATED COMPUTING MACHINE AND METHOD (~~Attorney Docket No. 1934-15-3~~). The synchronization interface **150** may also generate a SYNC signal to trigger the pipeline circuit **80** or to trigger another peer. In addition, the events from the input-event queue **124** also trigger the sequence manager **148** to activate the hardwired pipelines **74₁-74_n** as discussed below.

[91] Typically, a SYNC signal triggers a time-critical function but requires significant hardware resources; comparatively, an event typically triggers a non-time-critical function but requires significantly fewer hardware resources. As discussed in previously cited U.S. Patent App. Serial No. 10/683,932 entitled PIPELINE ACCELERATOR HAVING MULTIPLE PIPELINE UNITS AND RELATED COMPUTING MACHINE AND METHOD (~~Attorney Docket No. 1934-15-3~~), because a SYNC signal is routed directly from peer to peer, it can trigger a function more quickly than an event, which must make its way through, e.g., the pipeline bus **50** (**FIG. 3**), the input-data handler **120**, and the input-event queue **124**. But because they are separately routed, the SYNC signals require dedicated circuitry, such as routing lines, buffers, and the SYNC interface **150**, of the pipeline circuit **80**. Conversely, because they use the existing data-transfer infrastructure (e.g. the pipeline bus **50** and the input-data handler **120**), the events require only the dedicated input-event queue **124**. Consequently, designers tend to use events to trigger all but the most time-critical functions.

[117] Then, the output-data handler **126** retrieves the exception identifier from the exception manager **88** and sends the exception identifier to the host processor **42** (**FIG. 3**) as discussed in previously cited U.S. Patent App. Serial No. 10/684,053—entitled COMPUTING MACHINE HAVING IMPROVED COMPUTING ARCHITECTURE AND RELATED SYSTEM AND METHOD—~~(Attorney Docket No. 1934-12-3)~~.

Alternatively, if there are multiple possible destination peers, then the exception identifier can also include destination information from which the subscription manager **138** determines the destination peer or peers (e.g., the host processor **42** of **FIG. 3**) of the identifier. The output-data handler **126** then sends the identifier to the destination peer or peers via the industry-standard bus adapter **118** and the industry-standard bus interface **91**.